AR-002-079

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ELECTRONICS RESEARCH LABORATORY

JINDALEE PAPER NO. 147

INVESTIGATIONS OF LEXIDATA 3400 IMAGE PROCESSOR AND DIAGNOSTICS

S.P. TUCKER

SUMMARY

(U) In this paper an overview of the structure and use of the display hardware to be used in JINDALEE Stage B is given. Some diagnostic programs which test aspects of the hardware are also discussed.

POSTAL ADDRESS: Chief Superintendent, Electronics Research Laboratory,
Box 2151, GPO, Adelaide, South Australia, 5001.

TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF FIGURES

1. INTRODUCTION

(U) The display task for the JINDALEE B radar will be met by three Lexidata Image Processors, of which, two will be dedicated detection/tracking stations while the third will be primarily used by the radar controller . With a complex equipment configuration such as that of JINDALEE it is necessary to have confidence in the integrity of all equipments. To this end routine checks of all possible equipments are desirable if not mandatory.

(U) The purpose of this paper is to provide both an overview of the structure of the Lexidata hardware and its use, and to describe a number of programs that have been developed to check (automatically where possible) the main elements of the display hardware. Faults can thereby be localised to particular circuit boards, and in some instances, to specific integrated circuits.

(U) In section 2 a brief description of the Lexidata displays and supplied software is presented. The diagnostic programs are discussed in section 3.

2. DESCRIPTION OF THE LEXIDATA 3400 IMAGE PROCESSOR

(U) In this section the hardware comprising the Lexidata 3400 and the supplied software are discussed.

2.1 Hardware

(U) The Lexidata 3400 Image Processor is a raster scan colour display system designed to be driven by a host computer. Its major elements are a high speed microprocessor (80nsec cycle time), colour look up table (LUT), scroll/zoom controller, refreshed image memory and hardware cursor. The video output drives a high resolution colour monitor, while a keyboard and trackball constitute the input devices. The configuration of the Image Processors to be used in JINDALEE are shown in fig. 1.

2.1.1 Microprocessor

(U) The microprocessor is a 12 bit bi-polar device driven by an operating system stored in 2K of ROM and controls all input/output between the 3400 and the host. The functions supplied in the ROM code are FORTRAN callable and facilitate display initialisation, vector,cursor and character generation, scroll or zoom of the image and write/read of image data. A detailed description of the functions available in the operating system along with the microprocessor instruction set is contained in the User's Manual (ref. 1). The program ROM has a 1K RAM extension to allow user written functions to be appended to the operating system. These must be down line loaded from the host after every power up of the 3400.

(U) Associated with the program memory is a 256 word RAM of general purpose registers used by the microprocessor as utility storage.

(U) Coupled with the microprocessor is a Master Memory Controller which manages the refresh of the MOS image memory and the CRT. It also arbitrates on the use of the two ports of the image memory (section 2.1.2). By freeing the microprocessor of these tasks, more efficient and faster execution of the microcode is achieved.

2.1.2 Image Memory

(U) The image memory is a MOS array having two ports to allow update by the microprocessor and independent refresh of the CRT by the master memory controller. The memory is arranged as a 640x512 array of picture elements (pixels) with each pixel having 9 bits of intensity (i.e. 512 levels). Alternatively, the memory can be considered as 9 planes each having a resolution of 640x512 bits. By using a plane enable mask, each plane can be independently selected, and within each plane, each pixel can be addressed. The user can therefore define a channel as a group of planes (which are not necessarily contiguous). If more than one channel is defined then a number of independent images can be stored in the memory, and by judicious loading of the LUT (section 2.1.3), individual or combination of images can be displayed. The LUT must be selectively loaded since the plane enable mask only determines which planes are effectively connected to the microprocessor for read/write operations. In comparison, the number of planes connected to the LUT is independent of the plane enable mask, and is always nine. To display an image contained in a subset of the memory planes it is necessary to simulate a plane enable mask between the memory planes and the LUT. This is achieved by the manner in which the LUT is loaded (section 2.1.3). This technique will of course reduce the number of selectable intensities for each pixel to 2**m, where m is the number planes effectively connected to the LUT.

(U) The Lexidata 3400 will support up to 12 planes connected to the scroll/zoom controller and up to 4 planes non-zoomed. In the JINDALEE configuration there are 8 zoomed planes and 1 non-zoomed plane.

2.1.3 Video LUT

(U) The video LUT is a RAM of 4K bytes divided into four channels. It can be considered as a 4 channel colour map with the channels representing black/white (B/W) for a monochrome monitor and red, green, blue (RGB) for a colour monitor. Each channel has a 10 bit input address (i.e. 1024 locations) and an 8 bit output value. The number of memory planes that can be connected to the LUT is therefore restricted to ten. The memory address range of each LUT channel is given in Table 1.

(U) The LUT maps the image memory by using the intensity value of a given pixel as an offset address within each LUT channel. The contents of the channels at these addresses are output via D/A converters to the monitors. Figure 2 provides a functional description of the LUT. From this figure, an intensity of Z causes the contents of LUT addresses Z,1024+Z,2048+Z and 3072+Z to appear at the B/W,R,G and B outputs, respectively.

(U) The RGB channels collectively provide a 24 bit output representing 2**24 possible intensities, independent of the number of image memory planes. For example, if m memory planes are connected to the LUT then although only 2**m intensity levels are possible for each pixel, these levels can be mapped to anywhere within the range 0 to 2**24-1. The number of meaningful addresses within each LUT channel is 2**m.

(U) The use of the LUT offers a considerable advantage if, for instance, it is required to view a given image using various colour schemes. In this case it is only necessary to reload the LUT rather than rewrite the whole image. For example, an 8 plane image memory would require up to 320K bytes (depending on image density) to be rewritten, whereas the loading of the colour LUT involves writing only 2**8 bytes into each of the R,G and B channels, that is, a total of 768 bytes. A further advantage was referred to in the previous section when multiple images are stored in predefined memory channels and are selectively displayed by choice of the LUT contents. An example of a similar technique is the use of the non-zoomed plane as an overlay plane. In this case the non-zoomed plane represents the most significant bit of each pixel, so to function as an overlay plane all pixel values having the MSB set must produce the same output from the LUT. A typical LUT mapping to achieve this, is graphically described in fig. 3. Here all locations in the range 256 to 511 in each LUT channel have a constant value (within the channel). The remaining meaningful locations , 0 to 255, contain the LUT mapping for the other 8 planes.

(U) A FORTRAN subroutine is available to graphically examine the contents of the LUT. Using this subroutine the contents of the LUT are read back and a spectrum of all available colours is displayed on the monitor. A graphical description of each LUT channel is also plotted as LUT contents against LUT location. The LUT mapping of fig. 3 is similar to the graphical output obtained using this routine.

2.1.4 Scroll/Zoom Controller

(U) The scroll/zoom controller can control up to 12 memory planes, however in the JINDALEE configuration only 8 planes are connected to it. The controller allows the image produced on the CRT from these planes to be magnified, or scrolled in both x and y directions. A magnification factor of M causes each pixel to be displayed M times in both x and y directions on the CRT. The CRT image therefore corresponds to a region in the image memory having dimensions 640/M by 512/M pixels; this region is called the scroll/zoom field. The location of this field within the image memory is controlled by the scroll coordinates. These are the coordinates that identify the pixel which appears in the top left of the CRT image. the relationship between the coordinate systems of the image memory and the CRT image is explained in fig. 4. From fig. 4, the scroll coordinates which will cause the pixel located at (x,y) in the image memory to appear at location (X,Y) in the CRT image are (x-X/M,y-Y/M). The 3400, however, interprets these coordinates as unsigned positive integers with the x value being modulo 640 and the y value modulo 512. The modulo interpretation is equivalent to the image memory being circular in x and y. A general definition of the scroll coordinates is then (640+x-X/M, 512+y-Y/M).

(U) Blanking within or without a rectangular region of the CRT image is a feature of the scroll/zoom controller. To utilise this feature a rectangular region is defined in the CRT coordinates with blanking or no blanking specified. The remainder of the CRT image is either blanked or not blanked but in a complimentary sense. The blanking does not affect the contents of the image memory but effectively acts as a window overlayed on the CRT image. The blanking (or margin) function is independent of the scroll and zoom functions.

2.1.5 Hardware Cursor

(U) An hardware cursor generator is available as one of the functions of the Triple Option Board. The generator has two modes of operation, namely, a matrix cursor or full screen cross hairs. The matrix cursor is a 64 by 64 pixel array which can be loaded with any cursor design under software control. The cursor can be positioned under software control or by the trackball. An offset exists between the actual cursor position and the trackball coordinates due to harware timing considerations. This can be compensated when initialising the cursor (command DSCSL ref. 1) by defining an offset of (59,27) . The origins of the matrix cursor and the cross hairs are then set to be the top left element and the intersection point respectively.

(U) The triple option board also supplies three RS232 ports for communication with interactive devices. The trackball is connected to one of these ports.

2.2 Software

(U) The software supplied with the Lexidata 3400 consists of an Image Display Operating System (IDOS), an associated FORTRAN Interface Library, a Cross Assembler and a Device Driver for use with the RSX-11M operating system. The packages are described in the following sections, while their use and interaction is depicted in figs. 5 and 6.

2.2.1 IDOS

(U) The IDOS resides in a 2K ROM within the microprocessor and consists of a polling routine,a keyboard service routine, a trackball service routine and a group of function subroutines which can be executed by the 3400 microprocessor. A command from the host computer is sent as a variable length block of data in which the first word identifies the function to be executed while the remainder of the block is parameters or data for that function.

(U) A FORTRAN library enables commands to be issued from the host as simple FORTRAN subroutine calls; a description of all available functions and associated FORTRAN calls is available in ref. 1. Interpretation of the calls is done by the FORTRAN Interface Library at task build time, when the subroutine calls are converted to writes to the Lexidata of the required varible length data blocks.

(U) The polling routine, which is initiated on power up of the 3400, is described in fig. 7. The command decoder checks that the function code is legal and then, using a pointer table of subroutine addresses, jumps to the requested routine. The subroutine would then interrogate the interface and read all required parameters and data.

(U) A sample FORTRAN program is given in Appendix I along with the compilation and task build procedures.

2.2.2 Cross Assembler

(U) As described in 2.1.1 a 1K RAM, designated the Writable Control Store (WCS), is available to accept assembled user written programs from the host. The WCS is an extension of the 3400 program memory and has an address range of 4000 to 5777 (octal). The ROM occupies the address space 0 to 3777. Entry to the WCS code is gained by specifying a function code of 32. which effectively causes an unconditional jump to location 4000 where a command decoder is executed. This decoder and associated pointer table has a similar structure to that in the ROM code but allows recognition of WCS function codes and implements the appropriate subroutine calls within the WCS. Therefore, to reference a WCS function the normal variable length command block must be preceeded by an additional function code of 32 decimal. This variation of the command block structure should be reflected in the modifications that must be made to the FORTRAN Interface Library in order that WCS functions are recognised at task build time.

(U) To facilitate assembly of such programs, a Cross Assembler is supplied. This can be regarded as a library of macro definitions to enable the DEC assembler (MACRO) to assemble the Lexidata code. A description of the cross assembler is available in ref. 2. The object file from the assembly is unsuitable for direct loading into the WCS, consequently a program (LOAD.FTN) has been developed to read the object file, extract the assembly code and load it into the WCS of the selected device.

(U) The sequence of steps that must be performed to create a new function to be resident in the WCS and usable from FORTRAN is outlined in Appendix II.

2.2.3 Device Driver

The Lexidata Device Driver allows a user task running under RSX-11M to issue I/O requests to the Lexidata 3400. The supplied code can optionally be built as loadable or resident, and in addition contains conditional assembly code to enable a PDP11/70 version to be built.

(U) A number of limitations and faults were found with this package, namely, only one physical device was allowable (LX0), and system crashes were prevalent following a task abort. These shortcomings were rectified mainly by Mssrs. G.S.Brimble and M.S.Stevens. The driver will now recognise up to three Lexidata devices connected to one host (LX0,LX1,LX2), and will safely handle a task abort.

3. DIAGNOSTIC TEST PROGRAMS

(U) A suite of FORTRAN diagnostic programs has been developed to exercise various subsystems of the Lexidata image processor, namely , the video LUT,the scroll/zoom feature and the refresh memory. An additional program is available to facilitate adjustment of the RGB monitor. These test programs are available in two forms; a general form to which all parametric input is supplied from the user terminal, and a restricted version in which all parameters have been preset. This latter form has been developed for inclusion with the suite of tasks which can be invoked from the supervisor to be used in the JINDALEE Stage B radar (ref. 3). This supervisor does not allow terminal interaction with a task; instead it is provided with a menu language which has commands to prompt for, and to store, parameters in a resident common area. For this reason it was decided to minimise interaction and use preset parameters. In the descriptions that follow the values of the preset parameters are indicated. Appendix III gives examples of how to run each test program.

3.1 Scroll/Zoom Test - ZUMTST

(U) There is no readback capability for scroll/zoom memory addresses so comparative checking cannot be performed by the host computer. The scroll/zoom feature is therefore exercised in an elementary fashion by the user visually inspecting an image and adjudicating on its integrity.

(U) A grid is written in the non-zoomed memory plane and the top left corner filled. The intensity is selected such that the non zoomed plane is filled as well as some of the zoomed planes. This box is then incrementally scrolled down the diagonal of the grid at one second intervals. When scrolling occurs, the box will separate into two images of different intensities, one in the non zoomed plane and the other in the zoomed planes. Fig. 8 depicts the image after two incremental scrolls. At the completion of this sequence the top left corner of the grid is incrementally zoomed at one second intervals.

3.2 Look Up Table Test - LUTST

(U) The video look-up table (LUT) RAM is exercised in two stages. The first involves clearing all locations in the RAM, then reading them back channel by channel. Error counters for each memory chip are set up and all detected errors are accumulated in them. On completion of this stage an error map is printed at the line printer. The error map identifies each chip by its physical location on the LUT board, and indicates the number of errors in each chip. The second stage of the test involves repeating the first stage, but initially setting all LUT locations rather than clearing them.

(U) The only parameter required by this program is the number of memory planes connected to the LUT; however in the menu invoked version, this parameter will be set to nine.

(U) An example of the output obtained from this test is given in fig. 9. From this figure it is obvious that the memory chip located at position A16 on the LUT board of the Lexidata known as device LX0 is defective.

3.3 Memory Test - MEMTST

(U) The image memory consists of about 3 million bits arranged as an array of 640x512 picture elements (pixels), each having up to nine bits of intensity. To exhaustively test such a memory array would place an unacceptable overhead on a computing system primarily designed to control a radar. Since this memory test is designed to be run on a routine basis as part of an overall radar checkout, only a subset of the total memory will be exercised. Furthermore, the bits within this subset are tested independently,that is, the effect of a change of state of one bit on the state of its neighbours is not examined.

(U) Memory faults manifest themselves in two ways, either the image is mottled, or vertical lines which may or may not be continuous appear at 20 pixel spacing in the image. The mottled effect is due to individual bits within a chip failing, while the alternative symptom is due to total (or almost total) chip failure. Each memory plane of 640 by 512 pixels consists of twenty 16K by 1 bit RAM chips. The mapping between memory and CRT image is arranged such that each chip addresses 32 columns of 512 pixels spaced at 20 pixels. One row of the CRT image therefore addresses 32 bits from each chip. The effect of a total chip failure becomes obvious.

(U) For the purpose of the test the memory array is regarded as 9 planes of 512 rows with each row consisting of 640 elements. The test is conducted successively on each plane and within each plane , on each successive row. Initially the whole memory array is cleared and a plane selected. A row of '1's is written into and then read back from the memory. All errors are examined and the memory chips in which the error occurred are identified. The error counters representing the respective chips are then incremented. The same row of pixels is then filled with '0's and the process repeated. This sequence is repeated for each row in the plane and then for each plane in the memory. At the conclusion of the test an error map is printed which tabulates the number of errors against the physical location of the chip on the memory board.If all bits which have been tested from a given chip prove erroneous then a further map is printed indicating the faulty chips.

(U) It is emphasised that this test is not meant to be exhaustive, but to be indicative of error sources. Even though a memory chip may show errors in this test it is not conclusive that the chip itself is faulty. Apparent memory errors can be caused by bad contact at the edge connectors of the circuit boards or faults in address decoding logic. The former cause is obviously eliminated by re-inserting the boards into the backplane.

(U) A detailed error map can also be obtained which indicates the actual pixels that generated the errors; however, this is of limited use by virtue of the volume of the output. This map consists of one character per tested bit.

(U) Generally the test program MEMTST has as inputs, the range of planes and the range of rows to be tested. Experience has shown that a faulty chip generally produces faults at numerous bit locations. Therefore by only testing a subset of the rows the execution time of the test is reduced, while still testing a subset of the bit locations within each chip. The menu invoked version will test all planes but only 50 of the 512 rows of memory, thereby reducing the execution time by a factor of 10 to about one minute. The tabulated summaries are mandatory output.

3.4 Convergence Test - CONTST

(U) The high resolution monitor used in conjunction with the Lexidata Image Processor will require periodic adjustment of the beam convergence. If the output from the red,green and blue (RGB) electron guns are not correctly adjusted for convergence , a trio of images are visible. The convergence of the RGB outputs tends to drift with time and also depends on environmental factors such as temperature and physical orientation of the monitor with respect to the Earth's magnetic field. By displaying a suitable image on the monitor , the necessary adjustments can be made (ref. 4).

(U) A suitable image for this purpose is a grid pattern of equi-spaced lines or an array of equi-spaced points. Both of these patterns are generated with the test program CONTST. Generally the line or point spacing is user selectable; however the menu invoked version has a line spacing of 64 pixels and a point spacing of 32 pixels.

(U) The grid image generated by CONTST also facilitates correction of pin cushion effect, adjusting vertical linearity or when focussing the electron beams onto the monitor screen.

4. CONCLUSIONS

(U) A description of the major elements which comprise the 3400 Image Processor has been given with some emphasis placed on the role of the colour LUT. The LUT can be a powerful tool for image manipulation and some examples of its use were cited. A software description was presented with the objective of giving prospective users an overview and some insight into the software capabilities. Some procedures were given that facilitate modification of the IDOS and development of FORTRAN programs. A suite of diagnostic programs, which will be run on a routine basis during the course of JINDALEE B to verify the integrity of the display hardware, were described with examples of run procedures.

REFERENCES

No.	Author	Title
1	-	System 3400 Image and Graphics Processor User's Manual DC-1 Lexidata Corporation.
2	-	Lexidata Software Tape File` LEXMAC.TXT
3	A.M. Forbes	Users Guide to the JINDALEE Supervisor and the Writing of Menus. ERL-B144-TM (in publication)
4	-	BARCO Instruction Manual Section 6.

APPENDIX I

USING THE LEXIDATA 3400 WITH FORTRAN

(U) An example of a FORTRAN program with associated compilation and task build procedures.

```
C
C                 THIS EXAMPLE PROGRAM IS USED TO ILLUSTRATE THE
C                 COMPILATION AND TASK BUILD PROCEDURES FOR
C                 PROGRAMS THAT USE LEXIDATA SUBROUTINES.
C
C                 THIS PROGRAM PROMPTS FOR SOME PARAMETERS AND
C                 DRAWS A RECTANGLE.
C
          INTEGER X0,Y0,XLEN,YLEN,Z
          WRITE(5,5)
5         FORMAT('$ENTER LUN,LXDEV>')
          ACCEPT *,LUN,LXDEV             |INPUT LOGICAL UNIT NO. AND DEVICE NO.
          CALL    DSOPN(LUN,IERR,LXDEV)  |ATTACH DEVICE LXDEV TO LUN
          CALL    DSPLD(-1)              |INITIALISE IDOS
          CALL    DSCSL(10,0,0)          |SELECT MATRIX CURSOR
          CALL    DSCER                  |ERASE MATRIX CURSOR
          CALL    DSCLR(511)             |ERASE 9 PLANES
          CALL    DSLLU(1024,0,4096,0)   |ERASE LUT
          Z=10                           |SELECT INTENSITY
          CALL    DSLWT(1024+Z,1,255)    |LOAD LUT SUCH THAT Z=RED
          WRITE(5,6)
6         FORMAT('$ENTER RECTANGLE ORIGIN AND SIZE(X0,Y0,XLEN,YLEN>')
          ACCEPT *,X0,Y0,XLEN,YLEN
          CALL    DSVEC(X0,Y0,X0+XLEN,Y0,Z)        |DRAW RECTANGLE
          CALL    DSVEC(X0+XLEN,Y0,X0+XLEN,Y0+YLEN,Z)
          CALL    DSVEC(X0+XLEN,Y0+YLEN,X0,Y0+YLEN,Z)
          CALL    DSVEC(X0,Y0+YLEN,X0,Y0,Z)
          CALL    DSCLS                  |DETACH LXDEV
          END
```

```
Compile:
        >F4P RECT=RECT
Task build:
        >TKB RECT=RECT,[1,1]LEXLIB/LB,[1,1]F4POTS/LB
Run:
        >RUN RECT
         ENTER LUN,LX DEVICE NO.>1,1
         ENTER RECT. ORIGIN AND SIZE(X0,Y0,XLEN,YLEN)>50,50,400,200
```

(The absence of the prefix character '>' indicates a task generated prompt)

APPENDIX II

EXTENDING THE IDOS

(U) A listing of a user written WCS function is given in fig. I1. This function, DSROW, writes a row of pixels with each pixel having an arbitrary intensity. The pixels are only written in the image channel (see page 5.6 ref. 2) which must be defined as either 17 or 360 (octal); that is, the image channel must consist of either the first four or the last four memory planes respectively. To reduce I/O overheads the intensity values are input as bytes, that is, the intensity values are packed two per word.

(U) To retain compatibility with the ROM code the register assignments are made to symbolic names, so that the definition of these names is common to all code. Resolution of these symbolic names requires a further file (REG.MAC) to be linked at assembly. This file would also contain the command decoder cited in section 2.2.2 and all macros required by the cross assembler (ref. 3). In the following assembly, the file STOP.MAC contains the cross assembler macro to mark the end of the Lexidata code, and the .END macro to terminate the MACRO input files. Assembly is achieved by:

```
>MAC WCSCOD=LEXMAC,REG,DSROW,STOP
```

where the file LEXMAC.MAC is the cross assembler.

The assembled code is loaded into the WCS by running LOAD viz:

```
>RUN LOAD

ENTER INPUT FILE NAME>WCSCOD.OBJ

ENTER LUN,LX DEVICE NO.>1,1
```

(The absence of the prefix character '>' indicates a task generated prompt).

(U) In order to use the newly created function from FORTRAN, the fORTRAN Interface Library must be modified by adding a subroutine to set up the required function command block. A listing of an appropriate routine is given in fig. I2. The library is then rebuilt as follows:

```
>MAC LEXLIB=LEXLIB

>LBR [1,1]LEXLIB/CR:40.:::OBJ=LEXLIB
```

A FORTRAN program which uses the function can then be compiled and task built as in Appendix I.

FIGURE I1

```
;
;       ROUTINE TO WRITE A ROW OF PIXELS HAVING RANDOM INTENSITY.
;       THE PIXELS ARE ONLY WRIITEN IN THE IMAGE CHANNEL, AND
;       FURTHER, THE IMAGE CHANNEL MUST BE 17 OR 360(OCTAL).
;
;       FORTRAN CALL:
;               CALL    DSROW(NDOP,X,Y,BUF)
;                       NDOP =  NO. OF POINTS TO BE WRITTEN
;                       X,Y  =  ORIGIN OF PIXEL ROW
;                       BUF  =  BYTE ARRAY CONTAINING INTENSITIES
;
;
;               REGISTER DEFINITIONS FOR DSROW
;
        NDOP=ARG0L
        X=ARG1L
        Y=ARG2L
        CH17=TR3                ;MASK FOR MEMORY PARTITION(1ST 4 PLANES)
        CH360=TR12              ;MASK FOR MEMORY PARTITION(2ND 4 PLANES)
        LBYTE=237               ;REG FOR LOWER BYTE
        UBYTE=236               ;REG FOR UPPER BYTE
        BYTCNT=TR13             ;NO. OF BYTES PER WORD
;
;               REGISTER DEFINITIONS FOR ROUTINE BYTES
;               THESE REGISTERS ARE LOADED HERE FOR EFFICIENCY
;
        MSK1=TR1        ;MASK FOR LOWER BYTE
        MSK2=TR2        ;MASK FOR MS 4 BITS
        MSK3=TR3        ;MASK FOR LS 4 BITS AND MEMORY CHANNEL
;
;
;
        LABEL   ROW
        LOADC   ARGCNT  3-1
        CALL    232,RET1
        LOADC   MSK1,377        ;LOAD MASKS NEEDED BY
        LOADC   MSK2,7400       ;ROUTINE BYTES
        LOADC   MSK3,17
        LOADC   CH360,360
        BUSEN   MPOUT
        LDA     NDOP            ;GET NO. OF BYTES
        CLC                     ;AND CONVERT TO A
        SRE                     ;WORD COUNT
        STA     NDOP
        LDA     X               ;LOAD X AND Y ADDRESS REGS
        OTA     LDX
        LDA     Y
        OTA     LDY
        LDA     ICLSB           ;GET IMAGE CHANNEL
        OTA     LDPE            ;LOAD PLANE ENABLE REG
        CSE     CH17            ;IS IMAGE CHANNEL = 1ST 4 PLANES?
        JUMP    C360            ;NO
        JUMP    ROW1            ;YES,SO DECREMENT WORD COUNT FIRST
```

```
LABEL   NXTW1
LOADC   BYTCNT,2-1       ;LOAD LOOP COUNTER
LOADC   ARGPTR,LBYTE     ;AND POINTER TO BYTE REGISTERS
CALL    BYTES,RET1       ;GET A WORD AND DECODE INTO BYTES
BUSEN   MPOUT
LABEL   NXT1
LDA     @ARGPTR          ;GET A BYTE
RAR                      ;AND DIVIDE BY 16
RAR
RAR
RAR
OTA     LDWD             ;LOAD MEMORY LOAD REG
WAIT    P2AVAIL          ;WAIT FOR END OF MEMORY CYCLE
OTA     SWRITE           ;START MEMORY WRITE CYCLE
WAIT    P2AVAIL          ;WAIT FOR END OF MEMORY CYCLE
CLA
INA
OTA     MEMINC
DSN     ARGPTR           ;DECREMENT BYTE POINTER
DSN     BYTCNT           ;DECREMENT LOOP COUNTER
JUMP    NXT1             ;GET NEXT BYTE
LABEL   ROW1
DSN     NDOP             ;DECREMENT WORD COUNT
JUMP    NXTW1            ;GET NEXT WORD
JMPI    RET0             ;RETURN
LABEL   C360
CSE     CH360            ;IS CHANNEL = 2ND 4 PLANES?
JMPI    RET0     ;NO - SO RETURN - THIS WILL CAUSE 3400 TO HANG
JUMP    ROW2             ;YES,SO DECREMENT WORD COUNT FIRST
LABEL   NXTW2
LOADC   BYTCNT,1
LOADC   ARGPTR,LBYTE
CALL    BYTES,RET1
BUSEN   MPOUT
LABEL   NXT2
LDA     @ARGPTR
OTA     LDWD
WAIT    P2AVAIL
OTA     SWRITE
WAIT    P2AVAIL
CLA
INA
OTA     MEMINC
DSN     ARGPTR
NOP
DSN     BYTCNT
JUMP    NXT2
LABEL   ROW2
DSN     NDOP
JUMP    NXTW2
JMPI    RET0

;
;       SUBROUTINE USED BY DSROW
;
;       THIS ROUTINE READS A WORD FROM THE HOST AND UNPACKS
;         IT INTO TWO BYTES. THESE ARE PASSED TO THE CALLING
;         ROUTINE VIA REGISTERS LBYTE AND UBYTE(LOWER AND UPPER
;         BYTE RESPECTIVELY).
;
;
          MSK1=TR1
          MSK2=TR2
          MSK3=TR3
          UBYTE=236
          LBYTE=237
          LABEL    BYTES
          BUSEN    OUTBUF
          WAIT     OAVAIL
          LAL                      ;GET LS 12 BITS
          STA      TR0
          AND      MSK1            ;SELECT LS BYTE
          STA      LBYTE
          LDA      TR0
          AND      MSK2            ;SELECT MS 4 BITS
          STA      TR0
          LAM                      ;GET MS 4 BITS FROM HOST
          AND      MSK3            ;SELECT MS 4 BITS
          ADD      TR0             ;ADD REMNANT FROM LS 12 BITS
          RAL                      ;ROTATE MS 4 BITS INTO LS 4 BITS
          RAL
          RAL
          RAL
          STA      UBYTE
          OTA      OBSO
          JMPI     RET1
```

FIGURE I2

```
;
;FUNCTION DSROW  -  DISPLAY A ROW OF RANDOM INTENSITIES
;
;
;DISPLAYS A ROW OF PIXELS FROM ORIGIN (X,Y) WITH
;INTENSITIES STORED AS BYTES IN ARRAY BUF. THE NO. OF BYTES
;IS SENT AS FIRST ARGUMENT.
;
;FORTRAN CALL:
;                CALL    DSROW(NDOP,X,Y,BUF)
;                        NDOP  = NO. OF POINTS TO BE WRIITEN
;                        X     = ABSCISSA OF ORIGIN
;                        Y     = COORDINATE OF ORIGIN
;                        BUF   = BYTE ARRAY OF INTENSITIES
;
;
;
DSROW::  CMPB    (R5),#4          ;ARG LIST CHECK
         BGE     1$
         BPT                      ;NOT ENOUGH ARGS
1$:      MOV     #PMBLK,R3        ;BUFFER FOR PASSING COMMANDS
         MOV     R3,BADD
         MOV     #LXWCS,(R3)+     ;LXL006 - INVOKES WCS (RAM)
         MOV     #LXROW,(R3)+     ;FUNCTION CODE
         BIT     #1,     ARG1(R5)         ;TEST FOR ODD BYTE COUNT
         BEQ     2$               ;EVEN
         INC     @ARG1(R5)        ;ODD,SO INCREMENT
2$:      MOV     @ARG1(R5),(R3)+ ;NDOP
         MOV     @ARG2(R5),(R3)+ ;X
         MOV     @ARG3(R5),(R3)+ ;Y
         MOV     #12,BSIZE        ;BUFFER CONTAINS LXWCS,LXROW,ARG1 TO ARG3
         JSR     PC,LWRITE
         BCS     63$              ;GIVE UP IF QIO REJECTED
         TSTB    LXISB            ;ERROR CHECK
         BMI     63$              ;IF ERROR,GIVE UP NOW
         MOV     ARG4(R5),BADD
         MOV     @ARG1(R5),BSIZE ;COUNT
20$:     JSR     PC,LWRITE
63$:     RTS     PC
```

APPENDIX III

RUNNING THE DIAGNOSTIC PROGRAMS

(U) Example runs of all diagnostic programs where the absence of the prefix character '>' indicates the text is user task generated.

1. Scroll/Zoom Test - ZUMTST

```
        >RUN ZUMTST
        ENTER LUN,LX DEVICE NO.>1,1
        ENTER NO. OF PLANES>9
        LX1 OPEN ON LUN 1 WITH ERROR CODE 0
```

2. LUT Test - LUTST

```
        >RUN LUTST
        ENTER LUN,LX DEVICE NO.>1,1
        ENTER NO. OF PLANES>9
        LX1 OPEN ON LUN 1 WITH ERROR CODE 0
```

3. Memory Test - MEMTST

```
        >RUN MEMTST
        ENTER LUN,LX DEVICE NO.>1,1
        ENTER NO. OF PLANES>9
        LX1 OPEN ON LUN 1 WITH ERROR CODE 0
        DO YOU WANT SHORT-FORM ERROR SUMMARY ONLY?Y
        ENTER RANGE OF LINES TO TEST>300,350
        ENTER RANGE OF PLANES TO TEST>1,9
```

4. Convergence Test - CONTST

```
        >RUN CONTST
        ENTER LUN,LX DEVICE NO.>1,1
        ENTER NO. OF PLANES>9
        LX1 OPEN ON LUN 1 WITH ERROR CODE 0
        ENTER GRID SPACING>64
```

TABLE 1. MEMORY ADDRESS RANGE OF EACH LUT CHANNEL

ADDRESS RANGE	CHANNEL
0 - 1023	B/W
1024 - 2047	R
2048 - 3071	G
3072 - 4095	B

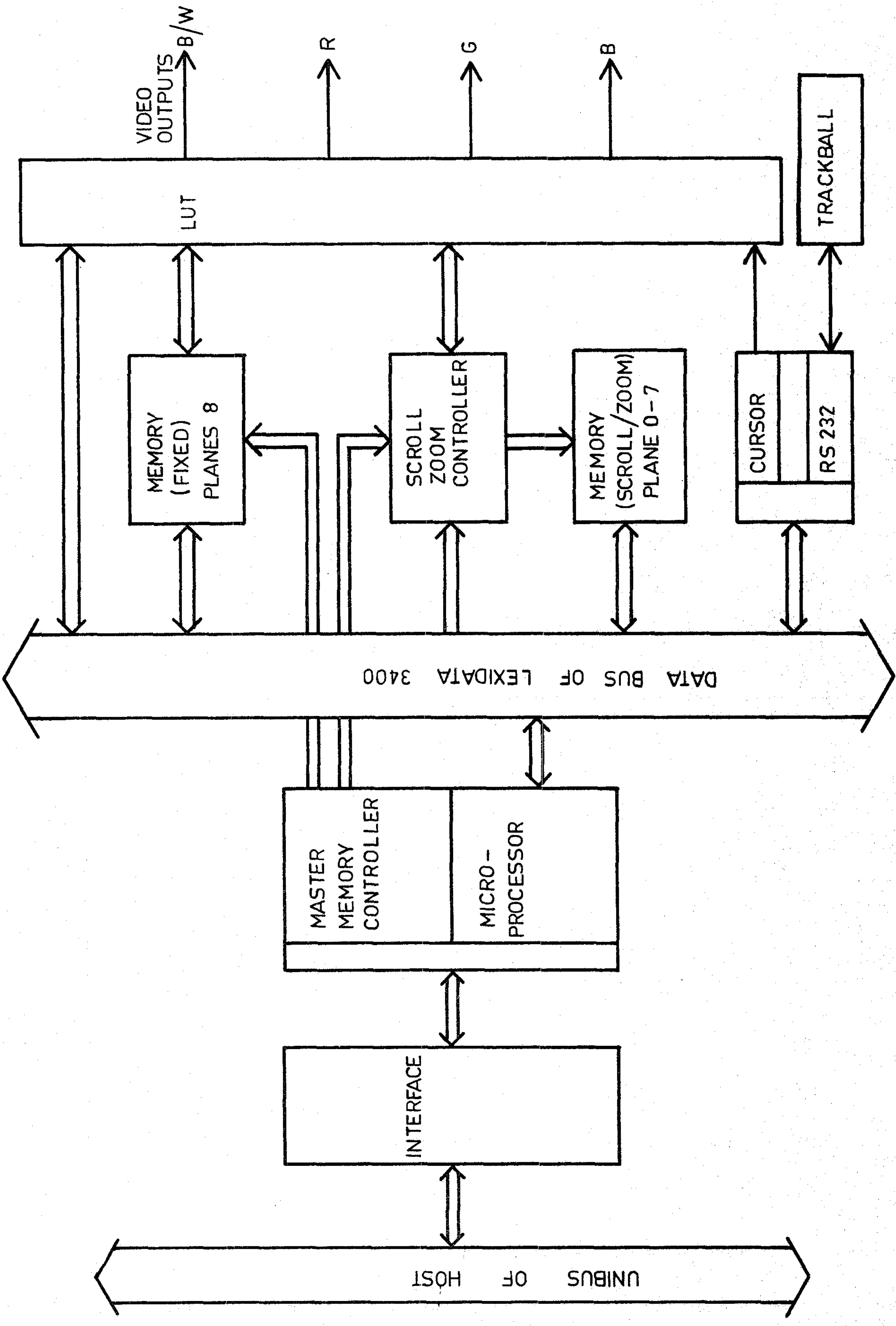


FIG. 1 BLOCK DIAGRAM OF LEXIDATA 3400 IMAGE PROCESSOR

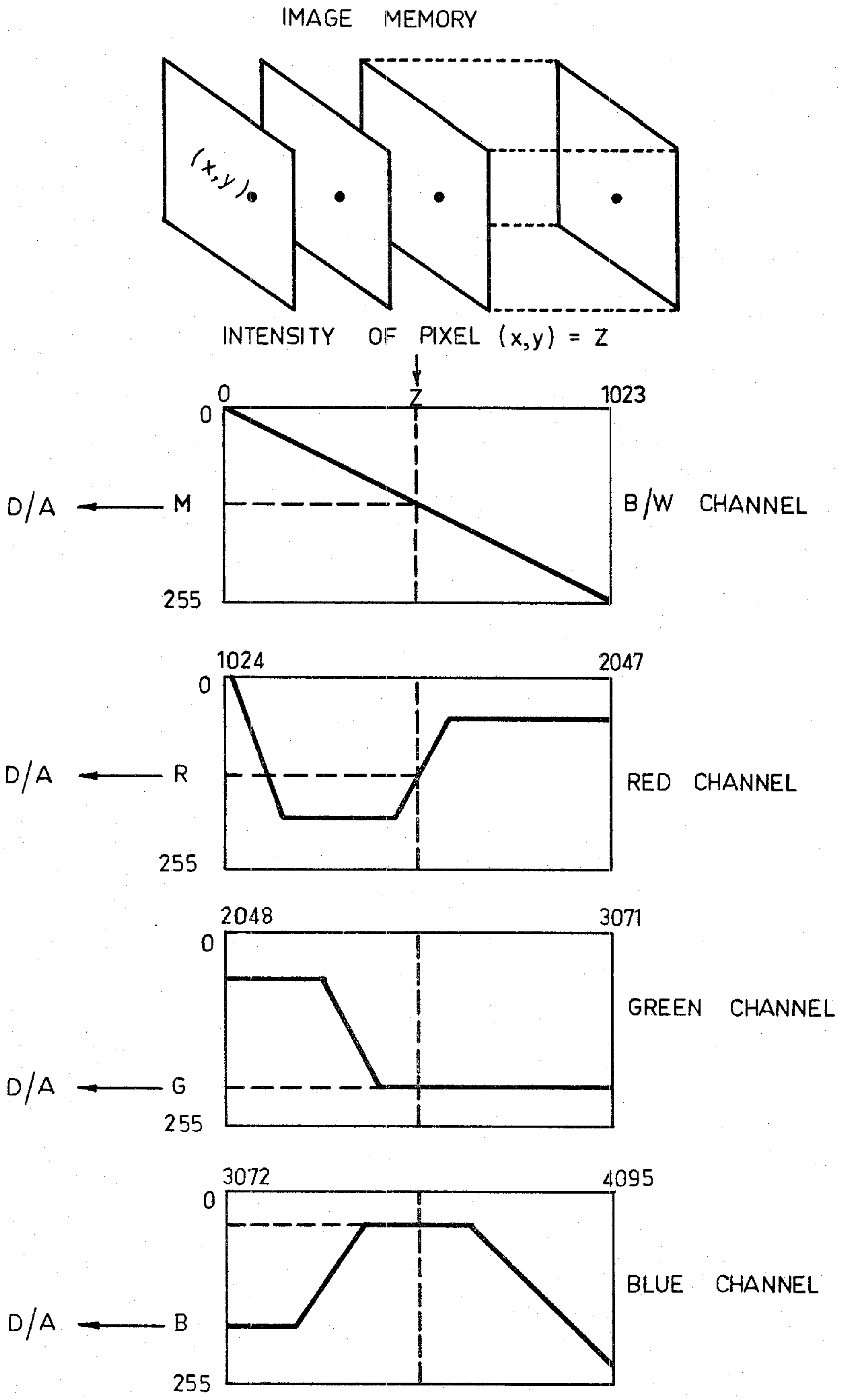


FIG. 2 FUNCTIONAL DESCRIPTION OF LUT

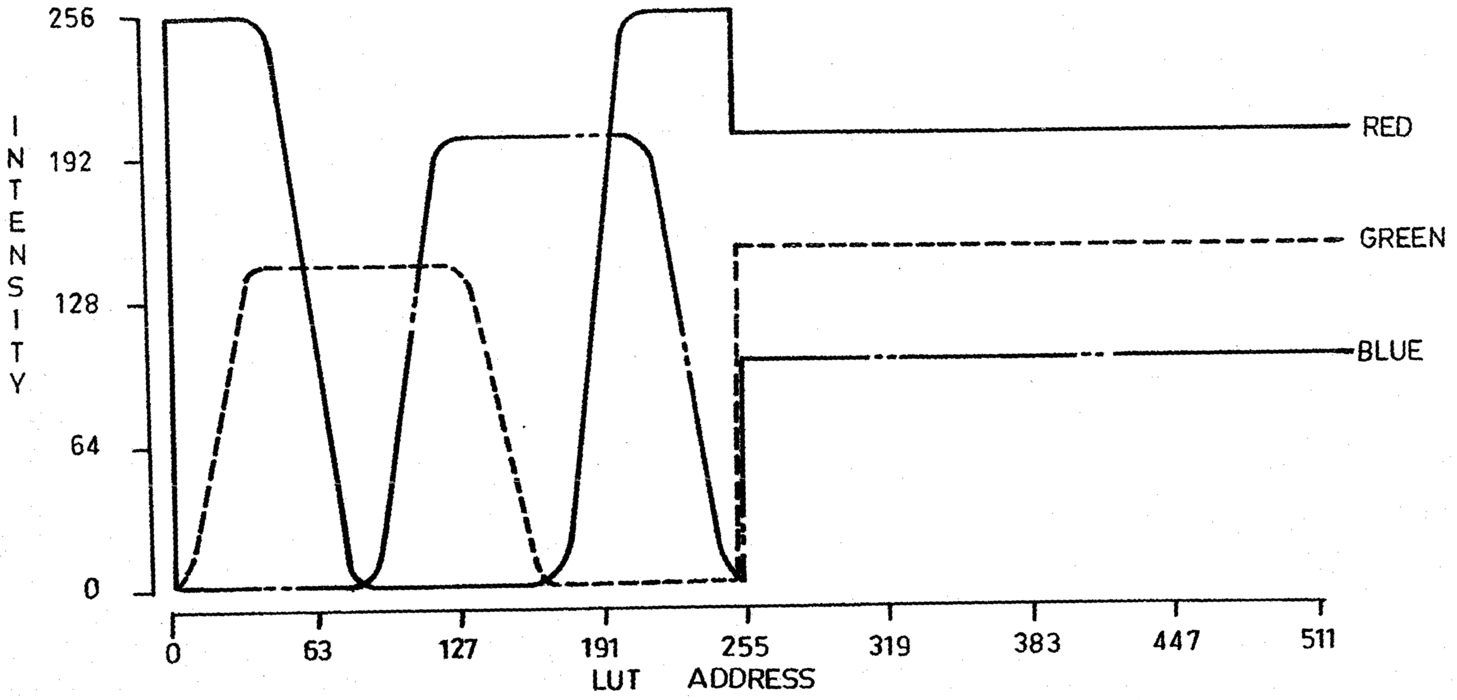


FIG. 3 LUT MAPPING FOR AN OVERLAY PLANE

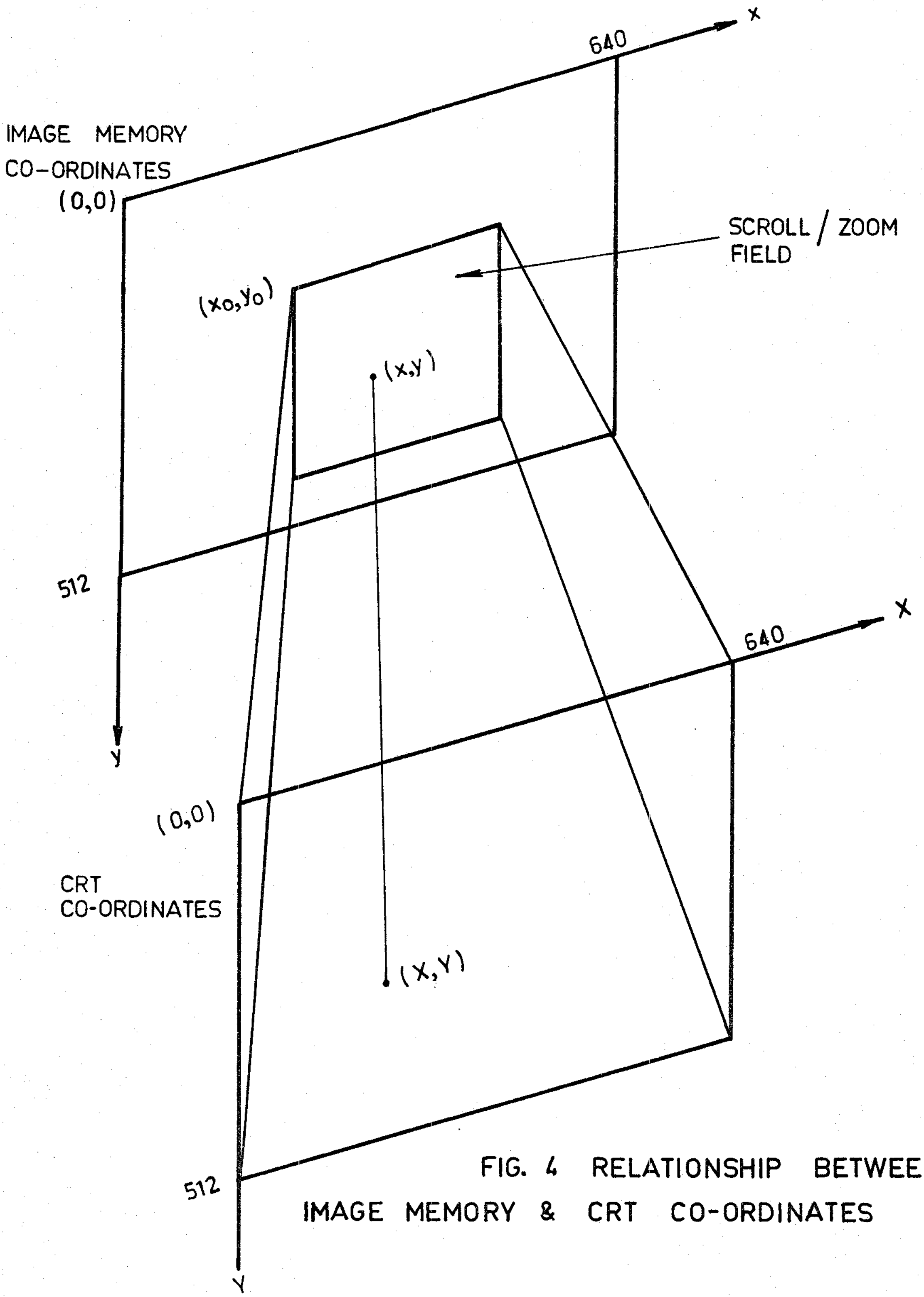


FIG. 4 RELATIONSHIP BETWEEN IMAGE MEMORY & CRT CO-ORDINATES

M = magnification factor (1, 2, ...16)

$(x - x_0)\ M = X$

Therefore $x_0 = x - \frac{X}{M}$

similarly $y_0 = y - \frac{Y}{M}$

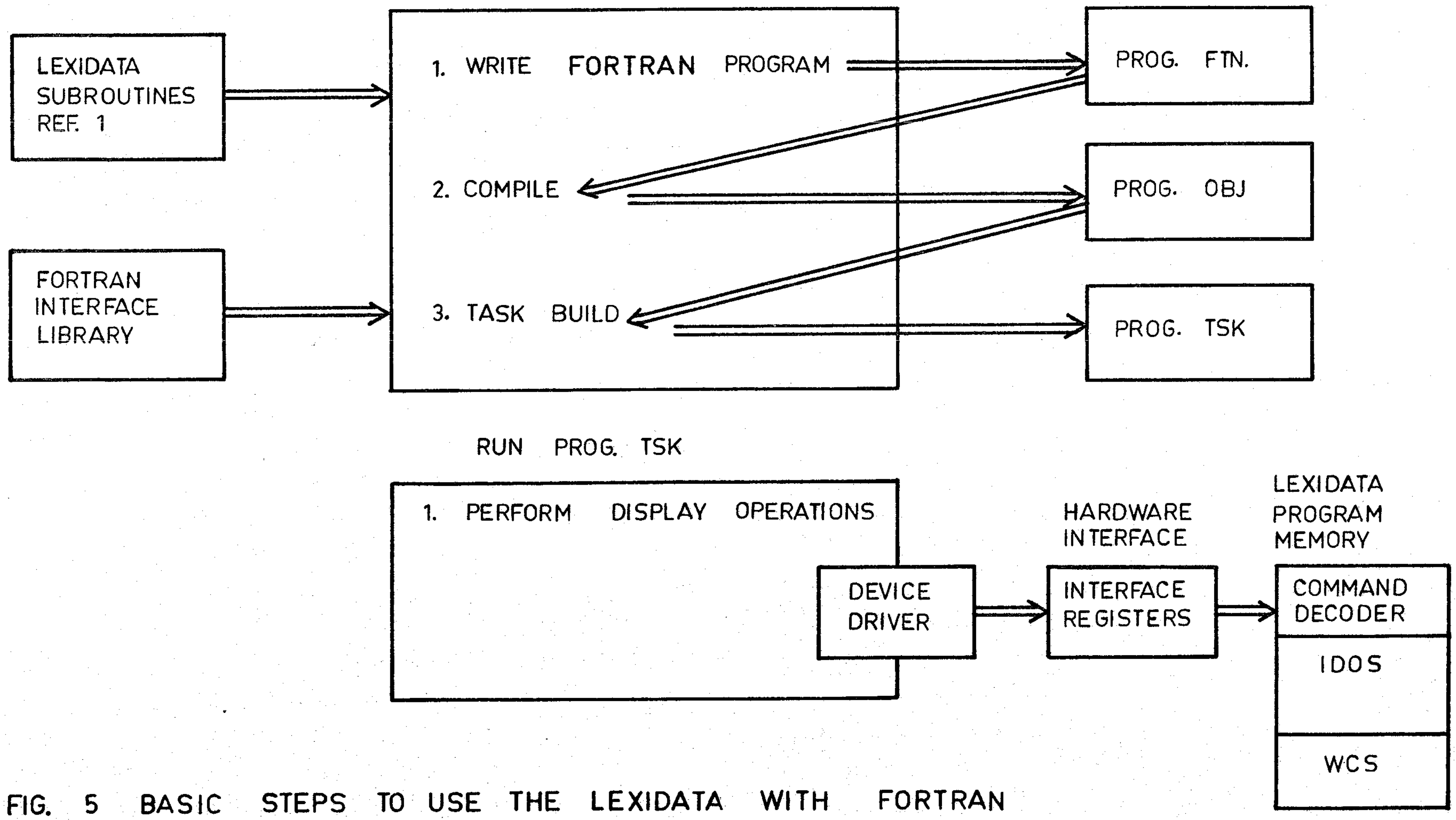


FIG. 5 BASIC STEPS TO USE THE LEXIDATA WITH FORTRAN

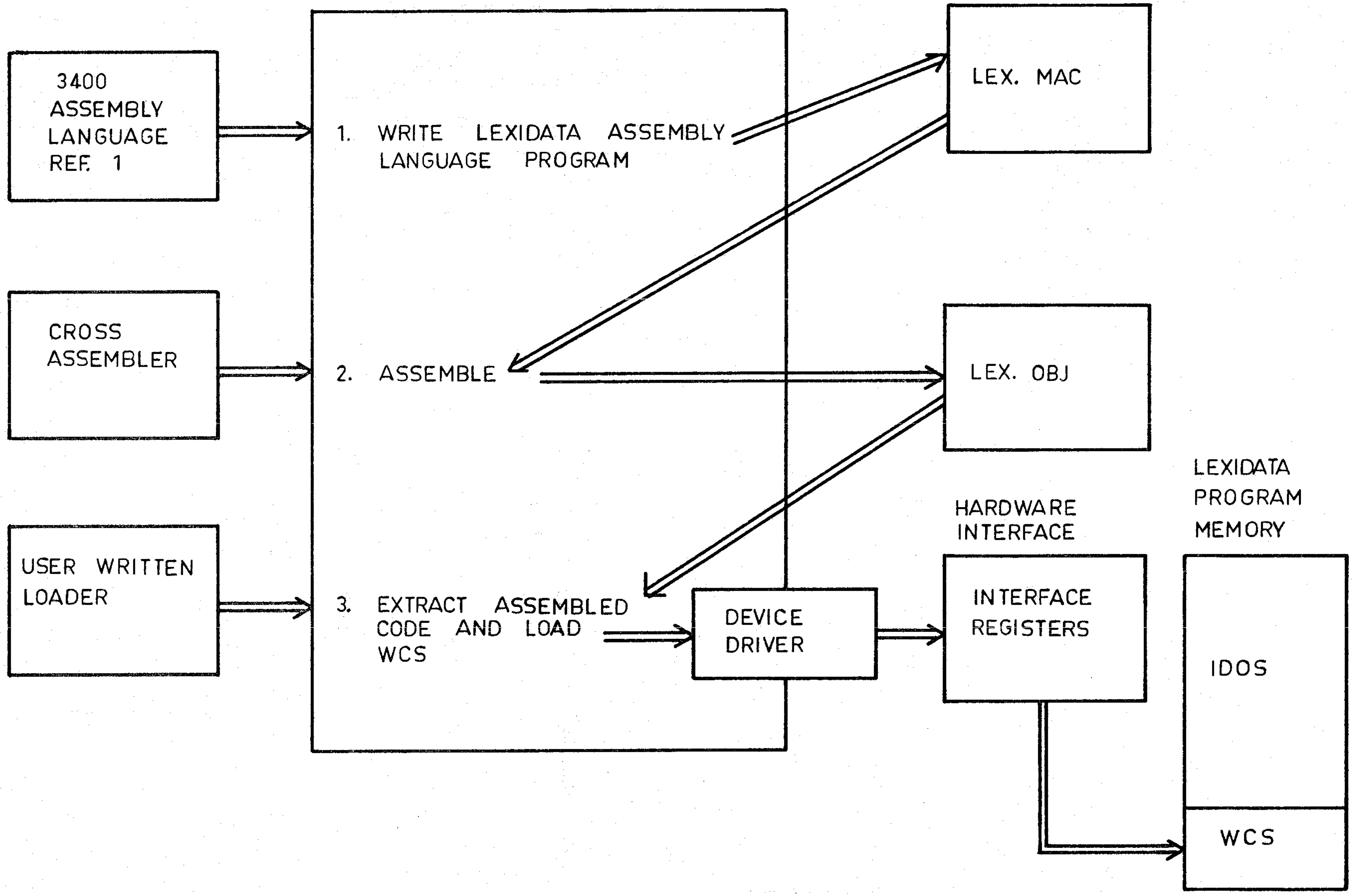
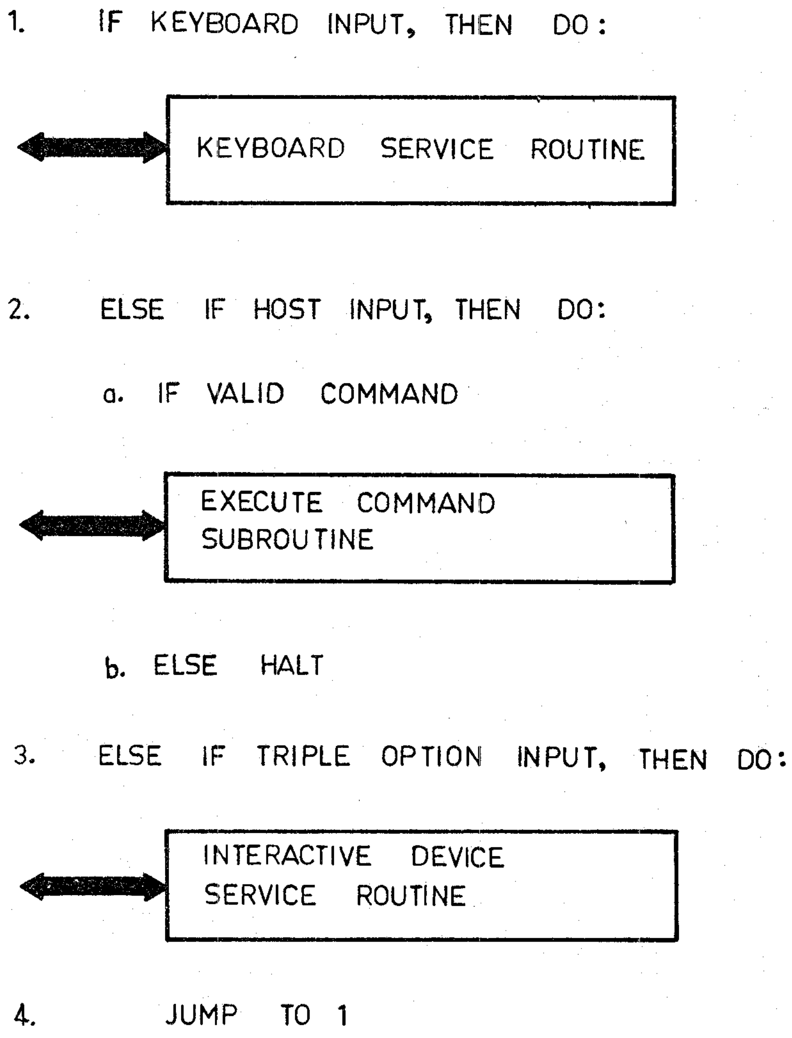
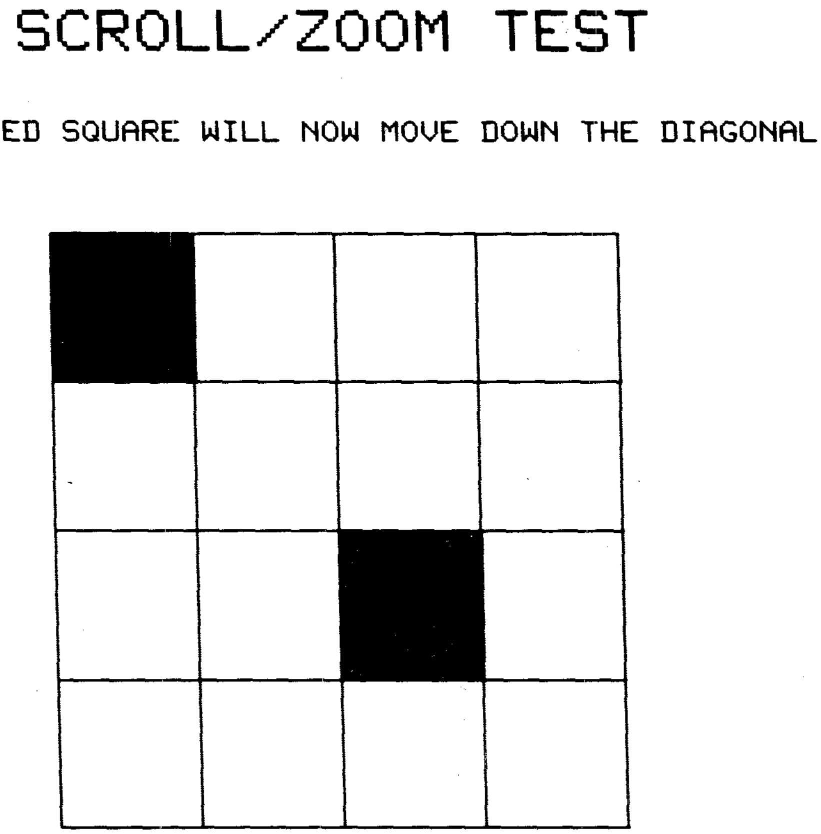


FIG. 6 BASIC STEPS FOR PROGRAMMING THE 3400 MICROPROCESSOR



```
1.    IF KEYBOARD INPUT, THEN DO:

          KEYBOARD SERVICE ROUTINE

2.    ELSE IF HOST INPUT, THEN DO:

      a. IF VALID COMMAND

          EXECUTE COMMAND
          SUBROUTINE

      b. ELSE HALT

3.    ELSE IF TRIPLE OPTION INPUT, THEN DO:

          INTERACTIVE DEVICE
          SERVICE ROUTINE

4.        JUMP TO 1
```

FIG. 7 POLLING ROUTINE



SCROLL/ZOOM TEST

COLOURED SQUARE WILL NOW MOVE DOWN THE DIAGONAL

FIG. 8 EXAMPLE OF SCROLL / ZOOM TEST OUTPUT

RESULTS OF LUT RAM TEST FOR LXO PERFORMED ON 3 NOVEMBER 1980
AT 16:53:03

SUMMARY OF CHIP ERRORS

The coordinates of the following tables are those of the actual chips on LUT board.

Table 1 : Obtained by writing then reading all zeros into the first 512 locations of each channel of the LUT.

Channel : B/W Red Green Blue

	A	B	C	D
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	62	0	0	0

Table 2 : Obtained by writing then reading all ones into the first 512 locations of each channel of the LUT.

Channel : B/W Red Green Blue

	A	B	C	D
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	255	0	0	0

FIG. 9 EXAMPLE OF LUT TEST OUTPUT

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	**2 SECURITY CLASSIFICATION**
AR Number: AR-002-079	a. Complete Document: UNCLASSIFIED ~~RESTRICTED~~
Report Number: ERL-B147-TM	b. Title in Isolation: UNCLASSIFIED
Other Numbers: -	c. Summary in Isolation: UNCLASSIFIED

3 TITLE

JINDALEE PAPER NO. 147

4 PERSONAL AUTHOR(S):	**5 DOCUMENT DATE:**
S.P. TUCKER	APRIL 1981
	6 6.1 TOTAL NUMBER OF PAGES 30
	6.2 NUMBER OF REFERENCES: 4
7 7.1 **CORPORATE AUTHOR(S):**	**8 REFERENCE NUMBERS**
Defence Research Centre Salisbury	a. Task: DEF 77/036
	b. Sponsoring Agency: Department of Defence
7.2 **DOCUMENT SERIES AND NUMBER** Electronics Research Laboratory B147-TM	**9 COST CODE:** 366965
10 IMPRINT (Publishing organisation)	**11 COMPUTER PROGRAM(S)** (Title(s) and language(s))
Defence Research Centre Salisbury	(i) Using the Lexidata 3400 with Fortran (ii) Extending the IDOS (iii) Running the diagnostic programs

12 RELEASE LIMITATIONS (of the document):

Distribution: Approved for Public Release ~~sts for this document must be referred to Chief Super Research Laboratory, Defence Research Centre, Salisb~~

12.0	OVERSEAS	NO	1	P.R.		A	5	B	X	C	X	D	X	E	X		

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

~~Not to~~ ~~TIS Current Aw~~areness Services ~~and~~
~~Catalo~~ ANNOUNCEABLE: to the Public

14 DESCRIPTORS:		15 COSATI CODES:
a. EJC Thesaurus Terms	Computer Graphics, Colour Displays, Image Processors	-
b. Non-Thesaurus Terms		

16 LIBRARY LOCATION CODES (for libraries listed in the distribution):

N/A

17 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

-

Security classification of this page: UNCLASSIFIED